

## LAB 8

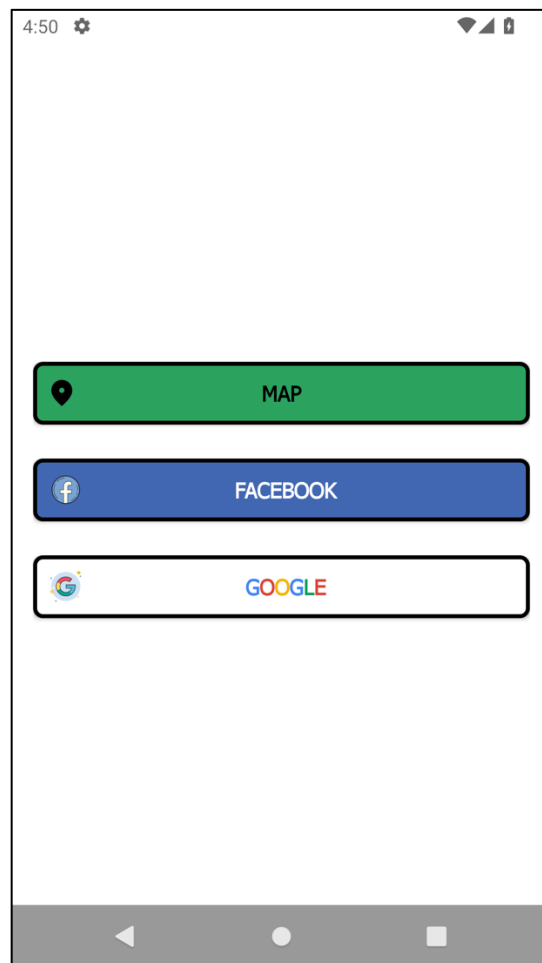
### MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Tích hợp và sử dụng Map vào dự án
- ✓ Đăng nhập với tài khoản facebook và google.

### NỘI DUNG

Xây dựng giao diện như hình bên dưới:



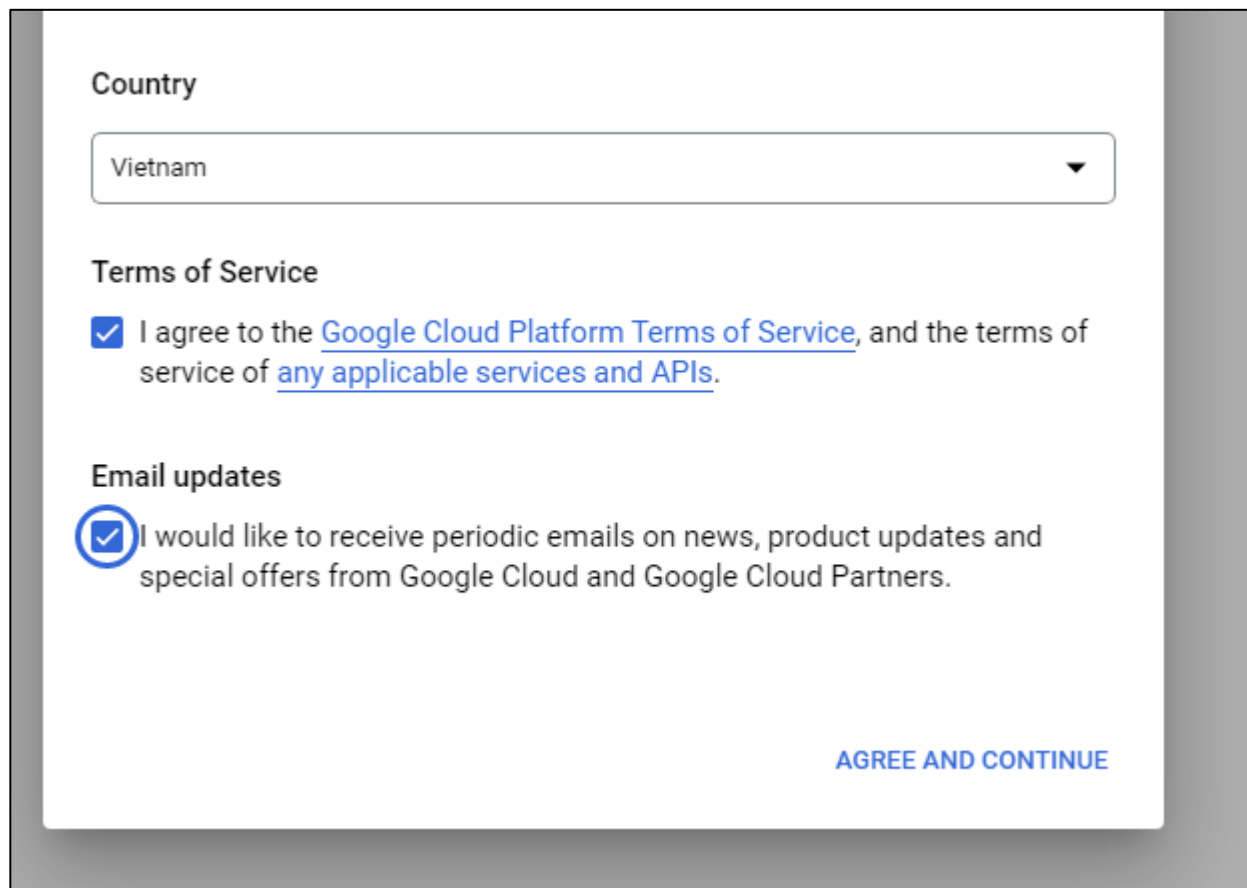
## BÀI 1: Tích hợp bản đồ vào dự án

Tài liệu tham khảo : [Devolopers](#)

### Bước 1: Truy cập vào trang web

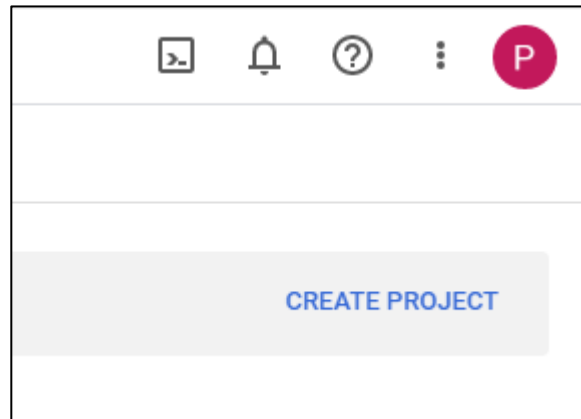
<https://console.cloud.google.com/projectselector2/google/maps-apis>

Sau đó đăng nhập bằng tài khoản google của mình. Sau khi đăng nhập hiện lên sẽ có một bảng thông tin như dưới, check hết các ô và nhấn **AGREE AND CONTINUE**






The screenshot shows a white dialog box with a grey border. At the top, it says "Country" above a dropdown menu showing "Vietnam". Below that is the "Terms of Service" section, which includes a checked checkbox and the text "I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#)." The "Email updates" section also has a checked checkbox and the text "I would like to receive periodic emails on news, product updates and special offers from Google Cloud and Google Cloud Partners." At the bottom right of the dialog box is a blue button labeled "AGREE AND CONTINUE".


Góc phải màn hình chọn vào **CREATE PROJECT**




Đặt tên cho project và nhấn **CREATE**

### New Project

 You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)   
[MANAGE QUOTAS](#) 

Project name \*  
My Map 

Project ID: my-map-383619. It cannot be changed later. [EDIT](#)

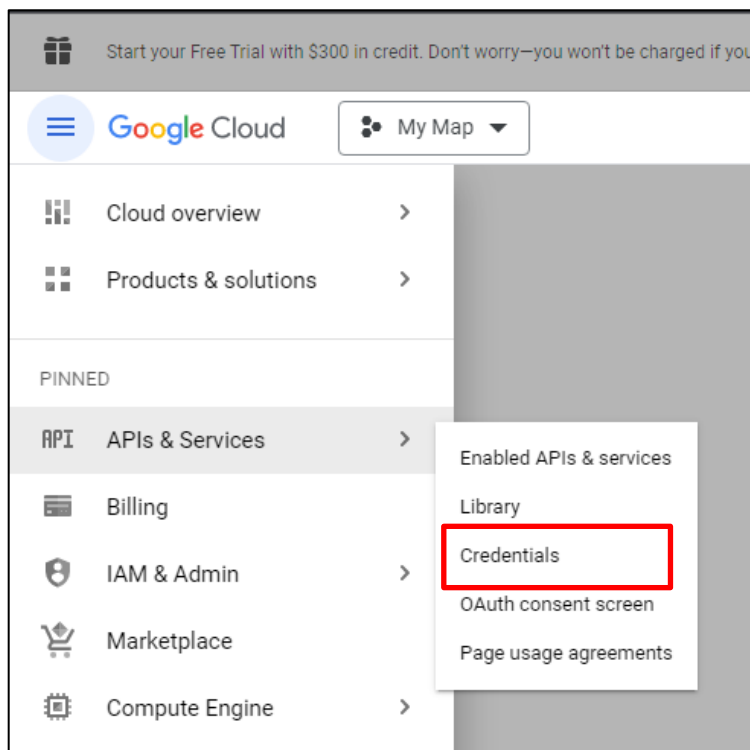
Location \*  
 No organization [BROWSE](#)

Parent organization or folder

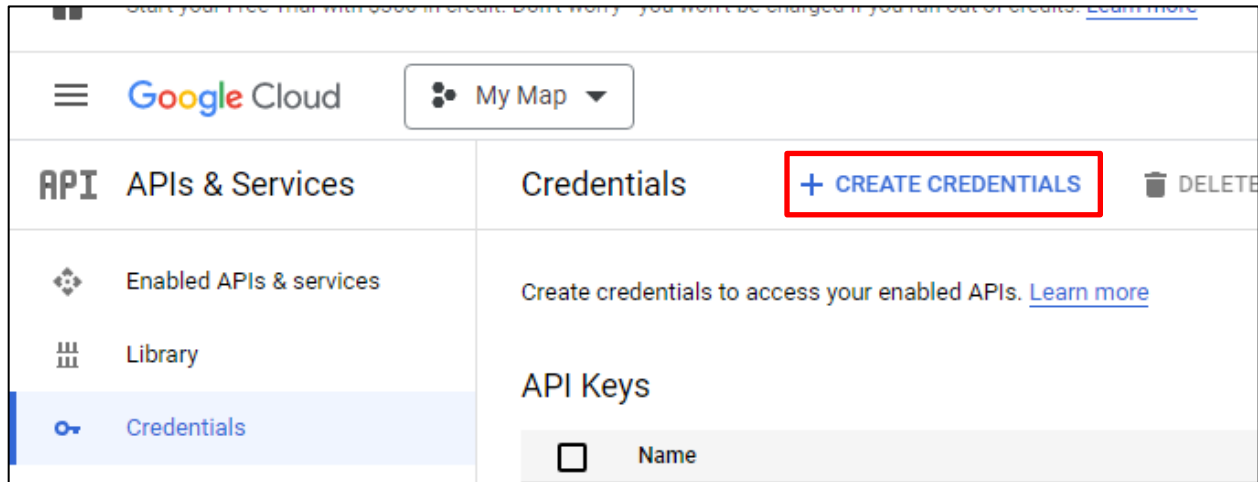
CREATE

CANCEL

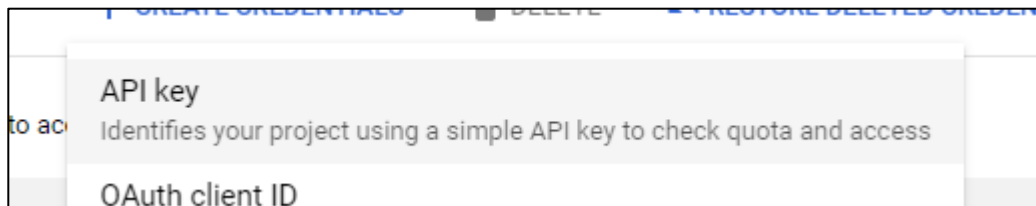
Sau khi tạo project xong, chúng ta truy cập vào trang <https://console.cloud.google.com/> chọn vào project vừa tạo ở trên. Trong mục **APIs & Services** chọn **Credentials**



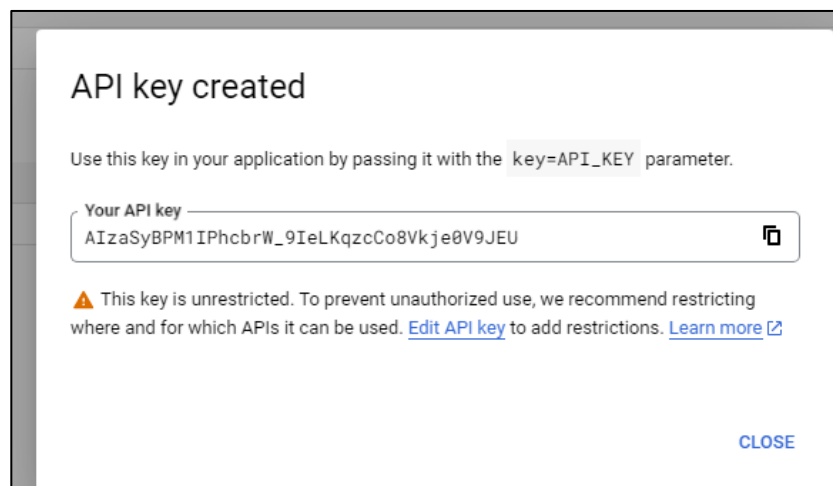
Sau đó chọn **CREATE CREDENTIALS**



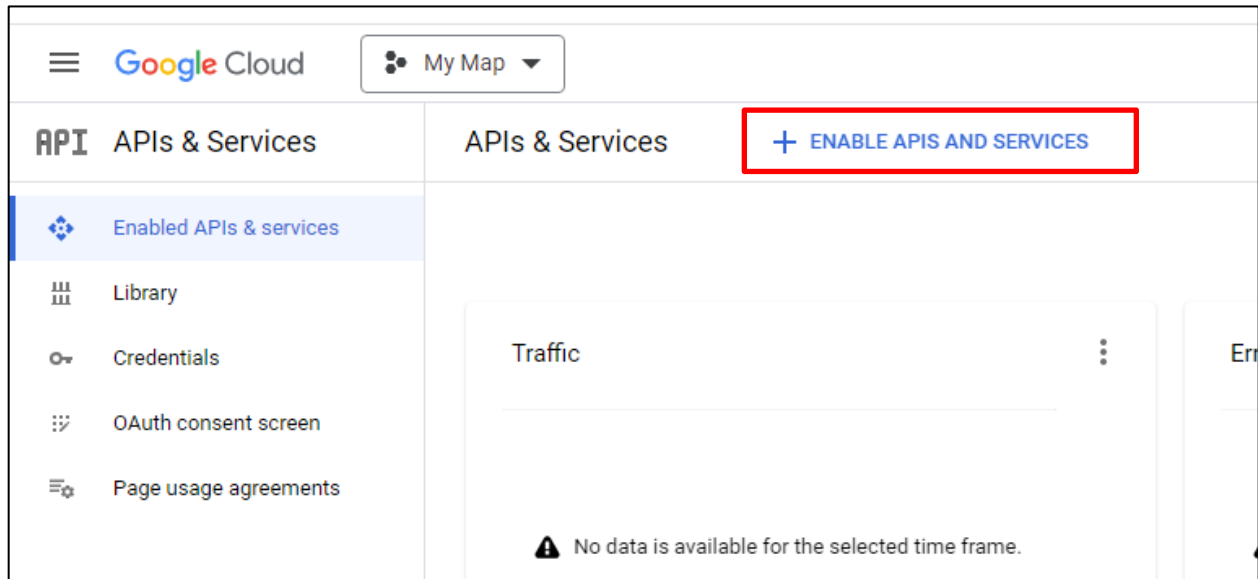
Chọn **API key**



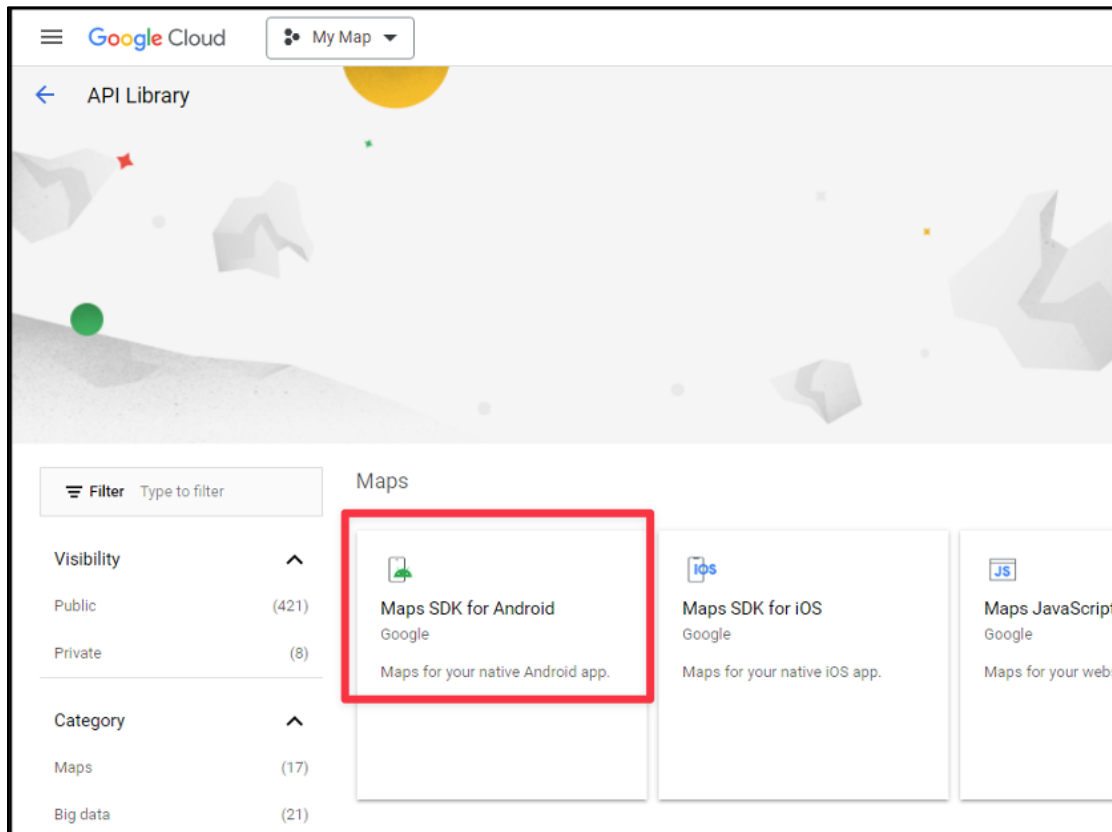
API key được tạo ra, copy lại key này, ở bước sau ta cần dùng đến



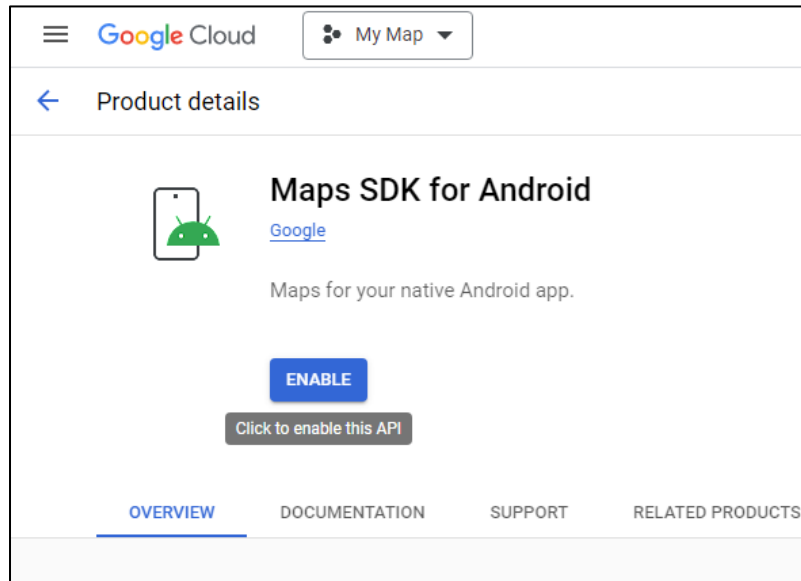
Tiếp theo chuyển qua tab **Enabled APIs & services**, chọn **Enable APIs & Services**



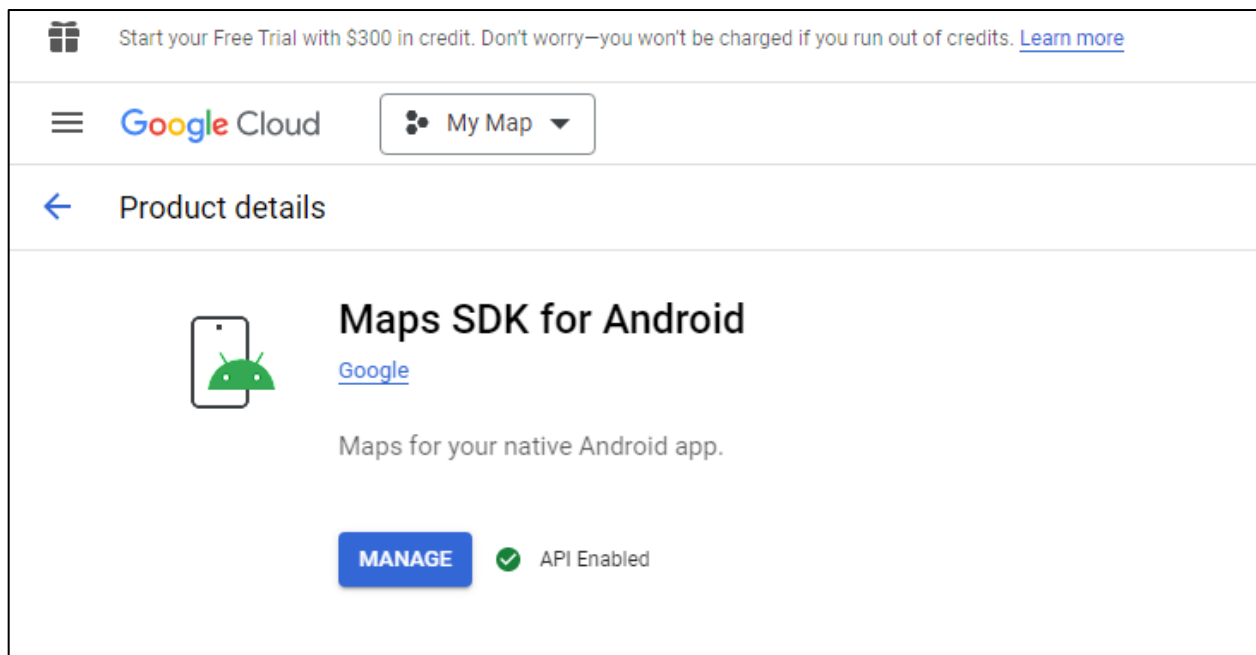
Chọn **Maps SDK for Android**



Chọn **Enable** Maps SDK for Android



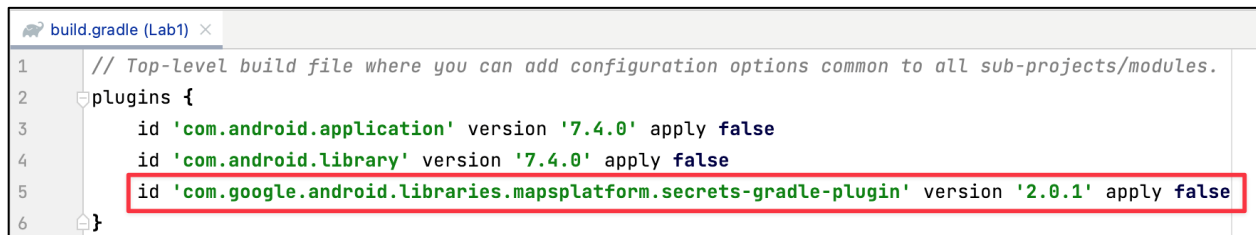
Sau khi Enable thành công, sẽ hiển thị kết quả như hình



**Bước 2:** Trong project android ta thực hiện setup các bước sau:

Ở file **build.gradle (Project)** ta thêm plugins

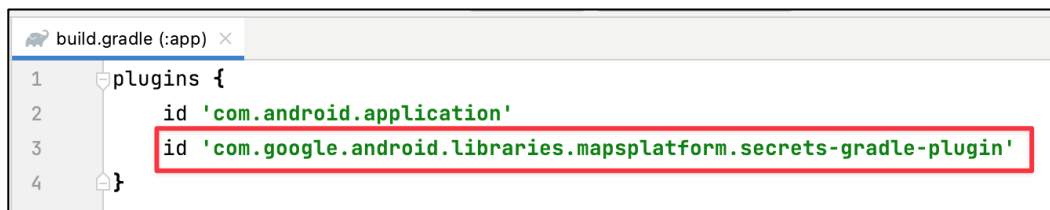
```
id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin' version '2.0.1' apply false
```



```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.
2 plugins {
3     id 'com.android.application' version '7.4.0' apply false
4     id 'com.android.library' version '7.4.0' apply false
5     id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin' version '2.0.1' apply false
6 }
```

Ở file **build.gradle (Module)** ta thêm plugins

```
id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'
```



```
1 plugins {
2     id 'com.android.application'
3     id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'
4 }
```

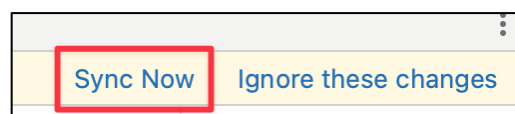
Tại **dependencies** trong **build.gradle (Module)** thêm thư viện:

```
implementation 'com.google.android.gms:play-services-maps:18.1.0'
```



```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.8.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    implementation 'com.google.android.gms:play-services-maps:18.1.0'
}
```

Sau đó nhấn **Sync Now**





Tại file **local.properties** ta thêm **MAPS\_API\_KEY** lấy ở bước 1

```

1  ## This file must *NOT* be checked into Version Control Systems,
2  # as it contains information specific to your local configuration.
3  #
4  # Location of the SDK. This is only used by Gradle.
5  # For customization when using a Version Control System, please read the
6  # header note.
7  #Mon May 29 01:36:14 ICT 2023
8  sdk.dir=/Users/dinhnguyenn/Library/Android/sdk
9  MAPS_API_KEY = AIzaSyDXC56nCpk9ndPDifKfBxuVIXBjeJgJvVM

```

Sau đó trong file **AndroidManifest** ta thêm thẻ meta-data vào application

```

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="${MAPS_API_KEY}" />

```

```

<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/Theme.Lab1"
    tools:targetApi="31">
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="${MAPS_API_KEY}" />

```

**Bước 3:** Tạo một Activity Map mới có layout như bên dưới

```

<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/map"
    tools:context=".Map"
    android:name="com.google.android.gms.maps.SupportMapFragment" />

```

Trong file java xử lý ta viết đoạn code sau:

```
public class Map extends AppCompatActivity implements OnMapReadyCallback {

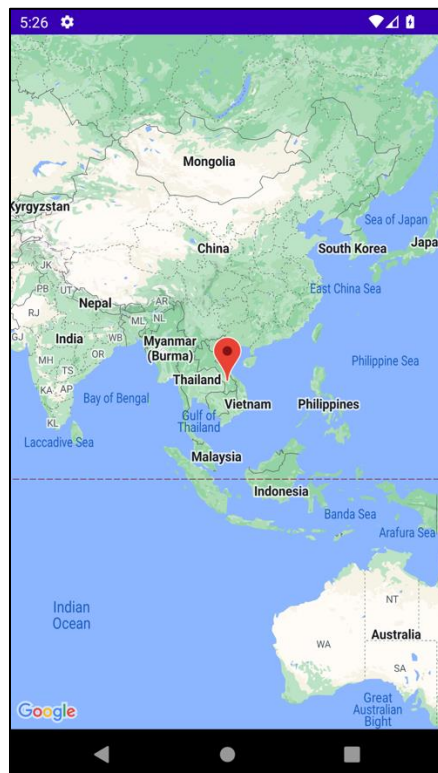
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map);
        SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(@NonNull GoogleMap googleMap) {

        // Add a marker in Sydney and move the camera
        LatLng FPT = new LatLng(16, 106);
        googleMap.addMarker(new MarkerOptions().position(FPT).title("FPT POLYTECHNIC"));
        googleMap.moveCamera(CameraUpdateFactory.newLatLng(FPT));
    }

    @Override
    public void onPointerCaptureChanged(boolean hasCapture) {
        super.onPointerCaptureChanged(hasCapture);
    }
}
```

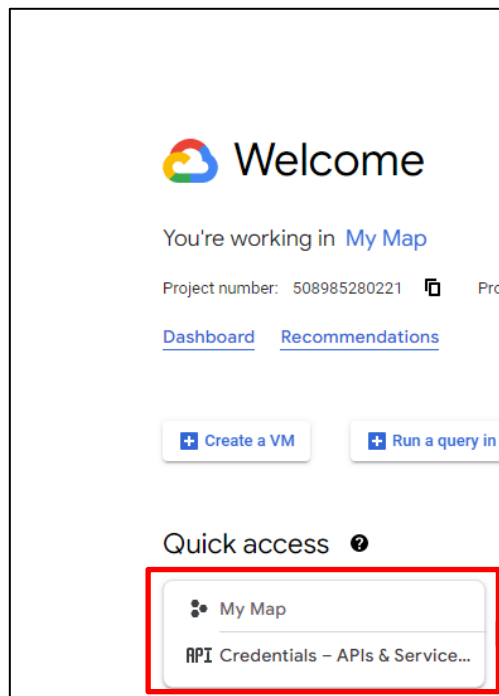
Chạy lại ứng dụng ta được kết quả:



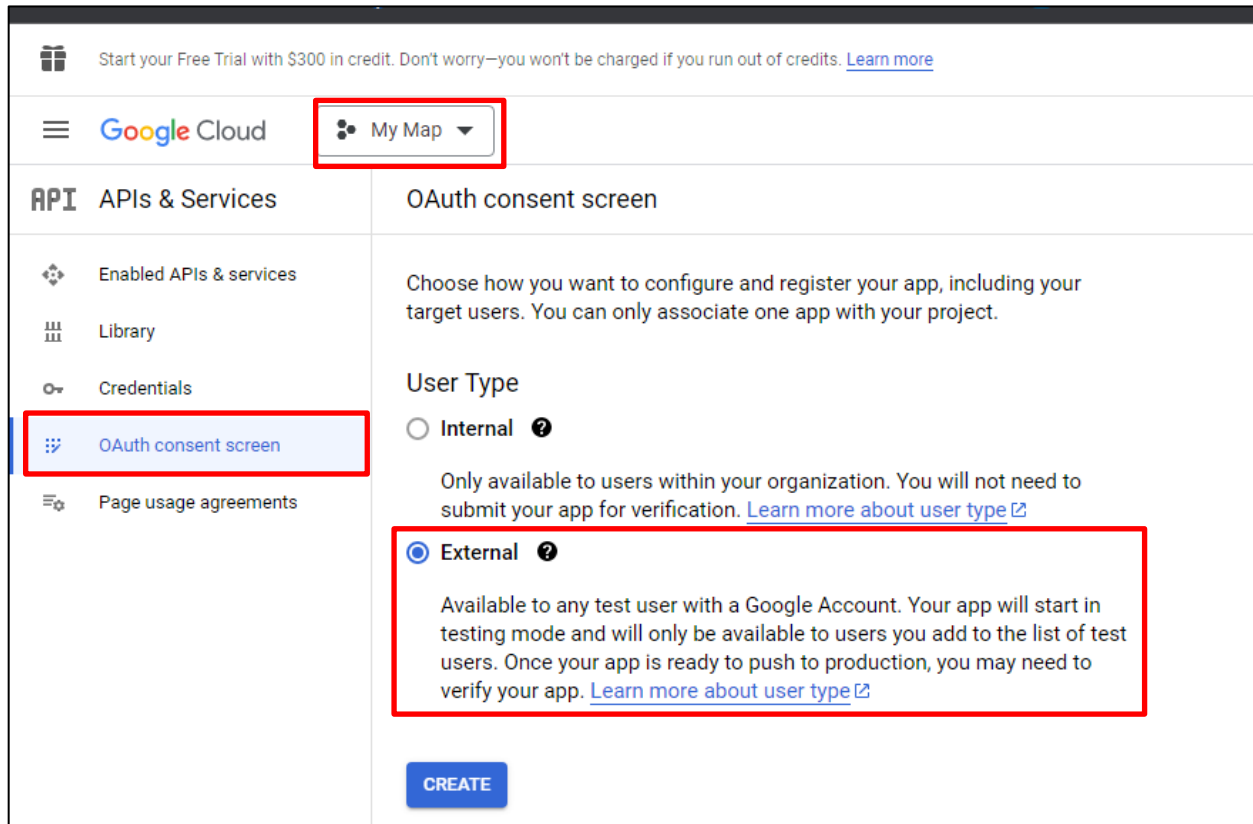
## BÀI 2: Thực hiện chức năng đăng nhập bằng tài khoản Google

**Bước 1:** Truy cập vào trang <https://console.cloud.google.com/>

Tạo project mới hoặc sử dụng lại project đã tạo ở Bài 01



Chọn vào **OAuth consent screen** → Chọn **External** → nhấn **Create**



Chúng ta sẽ điền các phần thông tin như sau:

- **App name:** Ta đặt theo tên của ứng dụng
- **User support email:** email ta dùng tạo project
- **Developer contact information:** email ta dùng tạo project

**App information**

This shows in the consent screen, and helps end users know who you are and contact you

**App name \***  
MyLoginz

The name of the app asking for consent

**User support email \***  
3@gmail.com

For users to contact you with questions about their consent

**Authorized domains**

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

+ ADD DOMAIN

**Developer contact information**

**Email addresses \***  
[redacted]@gmail.com

These email addresses are for Google to notify you about any changes to your project

Sau đó nhấn **Save And Continue**

**Developer contact information**

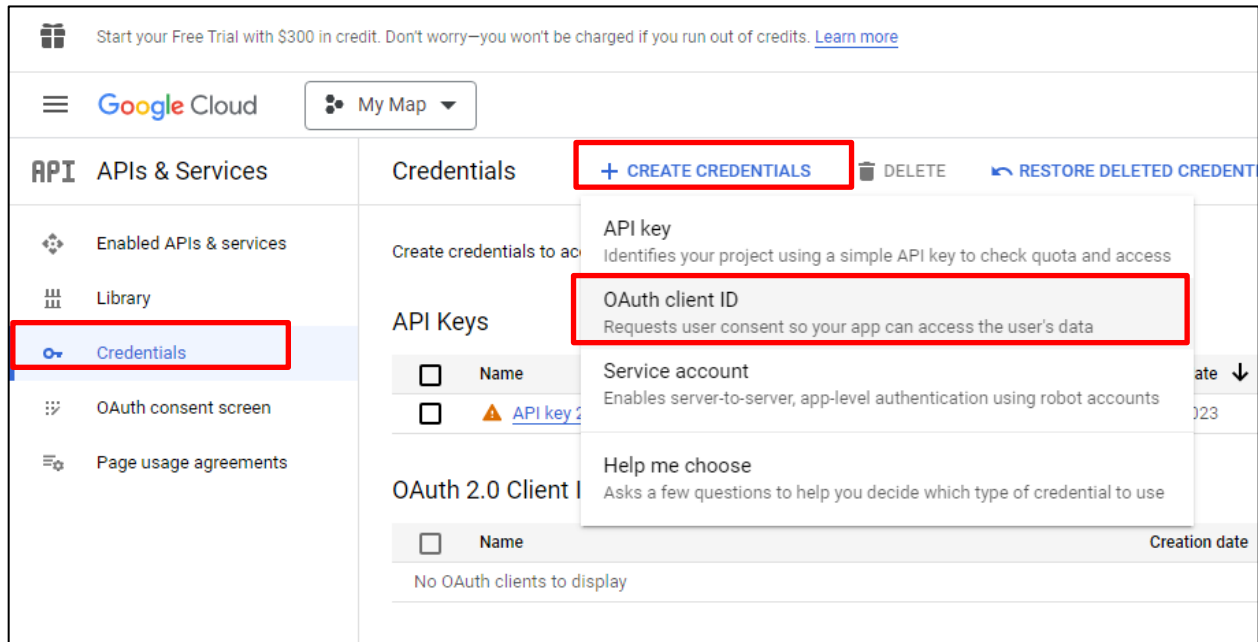
**Email addresses \***  
[redacted]@gmail.com

These email addresses are for Google to notify you about any changes to your project.

**SAVE AND CONTINUE** CANCEL

Ở các bước sau, ta không cần thiết lập gì cả nên chúng ta chuyển sang bước tiếp theo


Vào mục **Credentials** chọn **Create Credentials** → Chọn tiếp **OAuth client ID**



Ta sẽ điền các thông tin như sau

- **Application Type:** chọn **Android**
- **Name:** Tùy chọn hoặc có thể giữ nguyên
- **Package name:** tên package name của project
- **SHA1:** (cách lấy mã SHA1 ta có thể xem lại lab 7)

Sau đó nhấn **Create**

 **Create OAuth client ID**

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

**Application type \***  
Android

**Name \***  
Android client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

**Package name \***  
com.example.lab1

From your AndroidManifest.xml file.

**SHA-1 certificate fingerprint \***  
A9:16:E5:C1:E5:7[REDACTED]

SHA-1 signing certificate fingerprint restricts usage to your Android apps. [Learn more](#)

Use this command to get the fingerprint.

```
$ keytool -keystore path-to-debug-or-production-keystore -list
```

Note: It may take 5 minutes to a few hours for settings to take effect

**CREATE** CANCEL

Ta được kết quả như hình:

Credentials [+ CREATE CREDENTIALS](#) [DELETE](#) [RESTORE DELETED CREDENTIALS](#)

Create credentials to access your enabled APIs. [Learn more](#)

### API Keys

<input type="checkbox"/>	Name	Creation date ↓	Restrictions	Actions
<input type="checkbox"/>	<a href="#">API key 2</a>	Apr 14, 2023	None	<a href="#">SHOW KEY</a> ⋮

### OAuth 2.0 Client IDs

<input type="checkbox"/>	Name	Creation date ↓	Type	Client ID	Actions
<input type="checkbox"/>	<a href="#">Android client 1</a>	Apr 19, 2023	Android	508985280221-8sb1...	

### Service Accounts

[Manage service accounts](#)

<input type="checkbox"/>	Email	Name ↑	Actions
No service accounts to display			

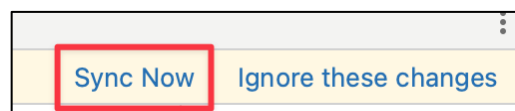
Quay lại project Android, trong **dependencies** ở file **build.gradle (Moule)** thêm thư viện: `implementation 'com.google.android.gms:play-services-auth:20.5.0'`

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.8.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

    implementation 'com.google.android.gms:play-services-maps:18.1.0'
    //Login with google
    implementation 'com.google.android.gms:play-services-auth:20.5.0'
}
```

Sau đó nhấn **Sync Now**





Trong Activity tạo một **ActivityResultLauncher** đặt tên là **registerForActivityResult**, **ActivityResultLauncher** này sẽ lấy được thông tin tài khoản mà người dùng đã đăng nhập

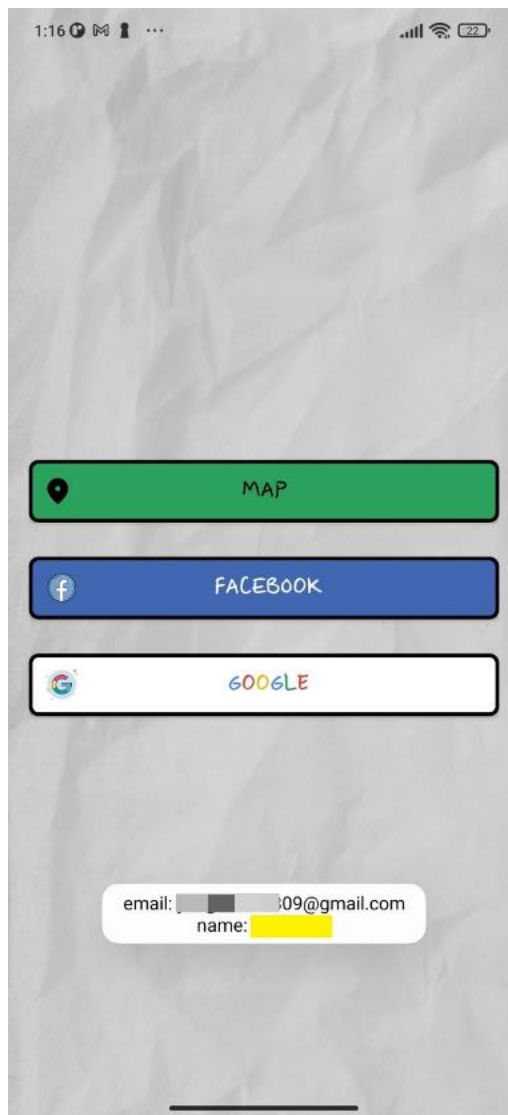
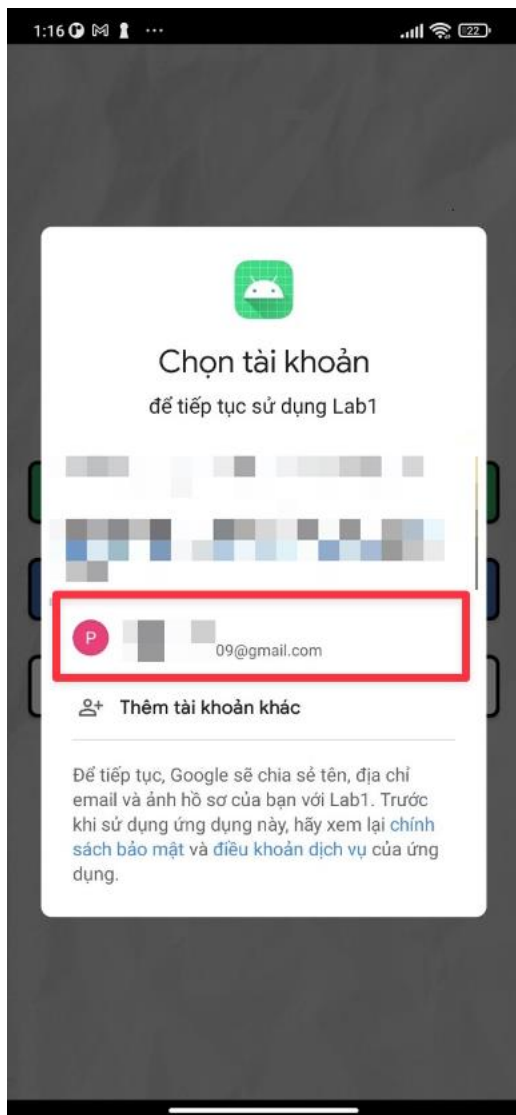
```
ActivityResultLauncher<Intent> registerForActivityResult
    = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            if (result.getResultCode() == Activity.RESULT_OK) {
                try {
                    Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(result.getData());
                    //Thông tin email
                    GoogleSignInAccount account = task.getResult(ApiException.class);
                    //Get email
                    String email = account.getEmail();
                    //Get name
                    String name = account.getDisplayName();
                    //Chúng ta có thể lấy ảnh của người dùng
                    //String avatar = account.getPhotoUrl().toString();
                    Toast.makeText(context: HomeActivity.this, text: "email: " + email + "\n" + "name: " + name, Toast.LENGTH_SHORT).show();
                } catch (Exception e) {
                    Log.e(TAG, msg: "onFailure: ", e);
                }
            } else {
                Toast.makeText(context: HomeActivity.this, text: "Thất bại", Toast.LENGTH_SHORT).show();
            }
        }
    });
```

Sau đó trong **onCreate()** viết đoạn code sau:

```
GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestEmail()
    .build();
mGoogleSignInClient = GoogleSignIn.getClient(activity: this, gso);

btnGoogle.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
        registerForActivityResult.launch(signInIntent);
    }
});
```

Ta được kết quả :



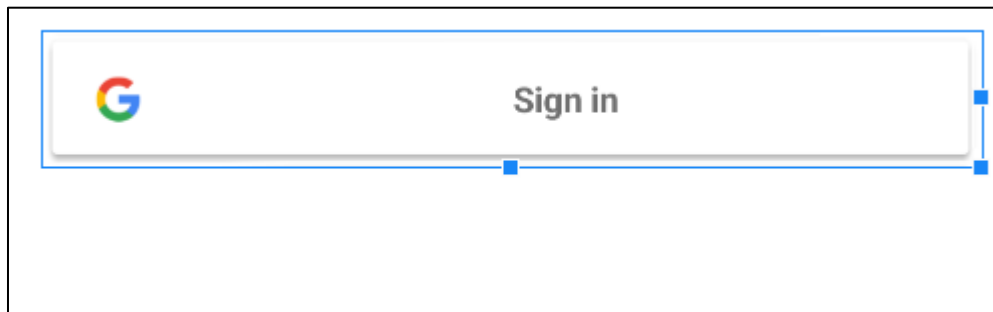
**Thông tin thêm:**

1. Ta có thể **đăng xuất** tài khoản bằng cách gọi hàm **signOut()** do GoogleSignInClient cung cấp

```
mGoogleSignInClient.signOut();
```

2. Ngoài cách tự tạo ra giao diện button cho login with google chúng ta có thể sử dụng giao diện button mà bên google thư viện google cung cấp

```
<com.google.android.gms.common.SignInButton  
    android:layout_marginTop="20dp"  
    android:id="@+id/sign_in_button"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```



3. Ngoài ra, khi đăng nhập thành công lần đầu tiên nếu ta không đăng xuất thì những lần sau đó **GoogleSignInClient** sẽ tự động đăng nhập tài khoản gần nhất mà ta đã đăng nhập trước đó. Nhờ thế ta có thể bắt sự kiện này để lưu trạng thái đăng nhập

```
@Override
protected void onStart() {
    super.onStart();

    GoogleSignInAccount account = GoogleSignIn.getLastSignedInAccount(this);

    if (account != null) {
        Toast.makeText(this, "Đã đăng nhập tài khoản " + account.getEmail(),
            Toast.LENGTH_SHORT).show();

        btnSignInButton.setVisibility(View.GONE);
        btnLogout.setVisibility(View.VISIBLE);
    }
}
```

## BÀI 3: Thực hiện chức năng đăng nhập bằng tài khoản Facebook

Truy cập vào trang <https://developers.facebook.com/docs/facebook-login/android>

### Bước 1: Chọn đăng nhập Facebook

Quyển  
Khắc phục sự cố  
Web  
Đối với thiết bị  
Guides  
Cách làm tốt nhất  
Thiết kế trải nghiệm người dùng

## Đăng nhập bằng Facebook dành cho Android - Bắt đầu nhanh

Facebook SDK dành cho Android cho phép mọi người đăng nhập vào ứng dụng qua phương thức Đăng nhập bằng Facebook. Khi mọi người đăng nhập vào ứng dụng bằng Facebook, họ có thể cấp quyền cho ứng dụng để bạn có thể truy xuất thông tin hoặc thực hiện hành động trên Facebook thay mặt họ.

Để biết dự án mẫu minh họa cách tích hợp phương thức Đăng nhập bằng Facebook vào ứng dụng Android, hãy xem FLoginSample trên GitHub.

Hãy làm theo các bước dưới đây để thêm phương thức Đăng nhập bằng Facebook vào ứng dụng của bạn.

### 1. Đăng nhập

Vui lòng đăng nhập Facebook để tạo ứng dụng hoặc đăng ký với tư cách nhà phát triển.

**Đăng nhập Facebook**

### 2. Tải xuống ứng dụng Facebook

5. Liên kết tên gói và lớp mặc định với ứng dụng của bạn  
6. Cung cấp hash khóa phát hành và phát triển cho ứng dụng  
7. Bắt đầu đăng nhập một lần cho ứng dụng của bạn  
8. Thêm nút Đăng nhập bằng Facebook  
9. Đăng ký gọi lại  
10. Kiểm tra trạng thái đăng nhập  
11. Bắt đầu đăng nhập tức thời  
Các bước tiếp theo

Sau khi Đăng nhập thành công sẽ hiển thị như hình bên dưới, tiếp tục **chọn tạo ứng dụng mới**

### 1. Chọn ứng dụng hoặc tạo ứng dụng mới

Chọn ứng dụng hoặc tạo ứng dụng mới để kéo thông tin về ứng dụng vào đoạn mã sau.

Tìm kiếm ứng dụng của bạn hoặc **Tạo ứng dụng mới**

### 2. Tải xuống ứng dụng Facebook

Tải xuống ứng dụng Facebook bằng cách nhấp vào nút bên dưới.

**Tải xuống Facebook dành cho Android**

Chọn thông tin loại ứng dụng sau đó nhấn **Tiếp**


Tạo ứng dụng


Loại


Chi tiết


Chọn loại ứng dụng


Sau khi tạo xong ứng dụng, bạn sẽ không thể thay đổi loại ứng dụng. [Tìm hiểu thêm](#)


 **Kinh doanh**  
Tạo hoặc quản lý các tài sản doanh nghiệp như Trang, Sự kiện, Nhóm, Quảng cáo, Messenger, WhatsApp và API Đồ thị trên Instagram bằng các quyền, tính năng và sản phẩm hiện có dành cho doanh nghiệp.


 **Người tiêu dùng**  
Kết nối ứng dụng của bạn với các sản phẩm và quyền dành cho người tiêu dùng, chẳng hạn như Đăng nhập bằng Facebook và Instagram Basic Display.

 **Trò chơi tức thì**  
Tạo game HTML5 được lưu trữ trên Facebook.

 **Chơi game**  
Kết nối một game ngoài nền tảng với phương thức Đăng nhập bằng Facebook.

 **Workplace**  
Create enterprise tools for Workplace from Meta.

 **Academic research**  
Connect to Facebook data and tooling to perform research on Facebook.

 **Không có**  
Create an app with combinations of consumer and business permissions and products.

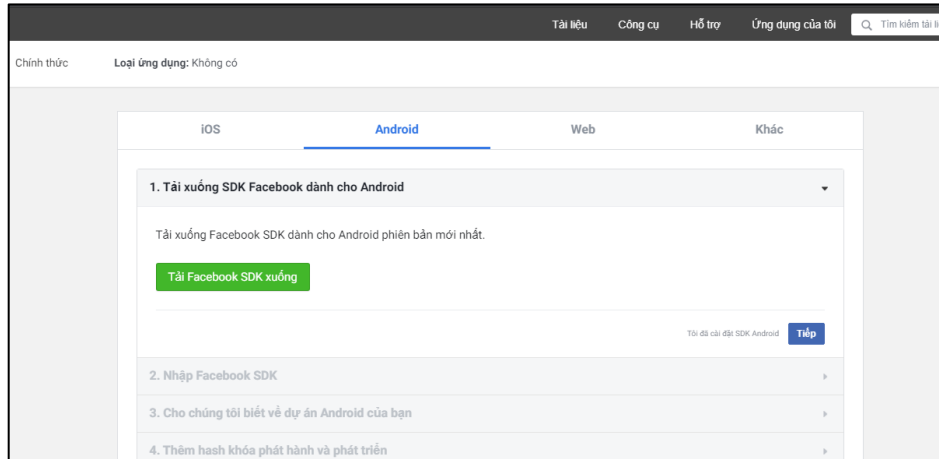
× Hủy

Tiếp

Điền thêm một số thông tin: tên ứng dụng, email liên hệ,... sau đó nhấn **Tạo ứng dụng**

Sau khi tạo ứng dụng xong, nhấn vào **Thiết lập Đăng nhập bằng Facebook**

Màn hình hiển thị như hình bên dưới, ta nhấn tiếp cho đến khi đến bước số 3



Điền thông tin **tên gói** và **tên lớp hoạt động** mặc định (Activity chạy lên mặc định khi ứng dụng được bật) sau đó nhấn **Save** và rồi nhấn **tiếp tục**

Ở đây các bạn lưu ý nếu chưa biết tên lớp hoạt động mặc định của project mình là Activity nào, bạn có thể truy cập **AndroidManifest.xml** để xem

3. Cho chúng tôi biết về dự án Android của bạn

**Tên gói**  
Tên gói xác định duy nhất ứng dụng Android của bạn. Chúng tôi sử dụng thông tin này để cho phép mọi người tải xuống ứng dụng của bạn từ Google Play nếu họ chưa cài đặt. Bạn có thể tìm thông tin này trong Tập kê khai Android hoặc tệp build.gradle của ứng dụng.

com.example.lab1

**Tên lớp hoạt động mặc định**  
Đây là tên lớp hoàn toàn đủ điều kiện của hoạt động xử lý liên kết sâu, chẳng hạn như com.example.app.DeepLinkingActivity. Chúng tôi sử dụng thông tin này khi liên kết sâu tới ứng dụng của bạn từ ứng dụng Facebook. Bạn cũng có thể tìm thông tin này trong tệp kê khai Android.

com.example.lab1.HomeActivity

Save

Quay lại Tiếp tục



## Tại bước 4, ta chỉ quan tâm dòng **Hash Khóa**

**Tạo hash khóa phát hành**

Bạn phải dùng khóa phát hành để ký dạng số cho các ứng dụng Android thì mới có thể tải các ứng dụng này lên cửa hàng. Để tạo hash khóa phát hành, hãy chạy lệnh sau trên Mac hoặc Windows để thay thế bí danh khóa phát hành và đường dẫn đến cửa hàng khóa:

```
keytool -exportcert -alias YOUR_RELEASE_KEY_ALIAS -keystore YOUR_RELEASE_KEY_PATH | openssl sha1 -bin
```

Sao chép mã

Quá trình này sẽ tạo một chuỗi gồm 28 ký tự để bạn sao chép và dán vào trường bên dưới. Đồng thời, hãy xem [tài liệu về Android](#) để biết cách ký ứng dụng của bạn.

**Hash khóa**

Ví dụ: nm0blrXpAM3cUsh

Để lấy được Hash khóa, ta thực hiện như sau:

- Lấy mã SHA-1 (xem lại lab 7)
- Truy cập vào trang web <https://base64.guru/converter/encode/hex>
- Paste mã SHA-1 vào ô **Hex\*** và nhấn **Convert Hex To Base64**

**Hex to Base64**

The "Hex to Base64" converter is a smart tool which is able to convert online Hex values into Base64 and give you the original data.

**Hex\***

A9: [Redacted]

**Convert Hex to Base64**

**Base64**

qRI [Redacted]

- Paste đoạn mã được convert ở ô Base64 vào ô Hash khóa, sau đó nhấn **Save** rồi nhấn **Tiếp tục**

để biết cách ký ứng dụng của bạn.

Hash khóa

Save

Quay lại

Tiếp tục

Quay lại project android, ta thêm thư viện:

```
implementation 'com.facebook.android:facebook-login:latest.release'
```

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.8.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

    implementation 'com.google.android.gms:play-services-maps:18.1.0'
    //Login with google
    implementation 'com.google.android.gms:play-services-auth:20.5.0'

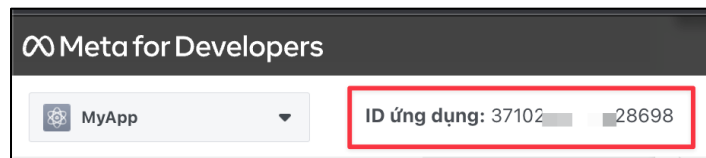
    //Login with Facebook
    implementation 'com.facebook.android:facebook-login:latest.release'
}
```

Trong file **strings.xml** của dự án ta thiết lập một số thông tin sau:

```

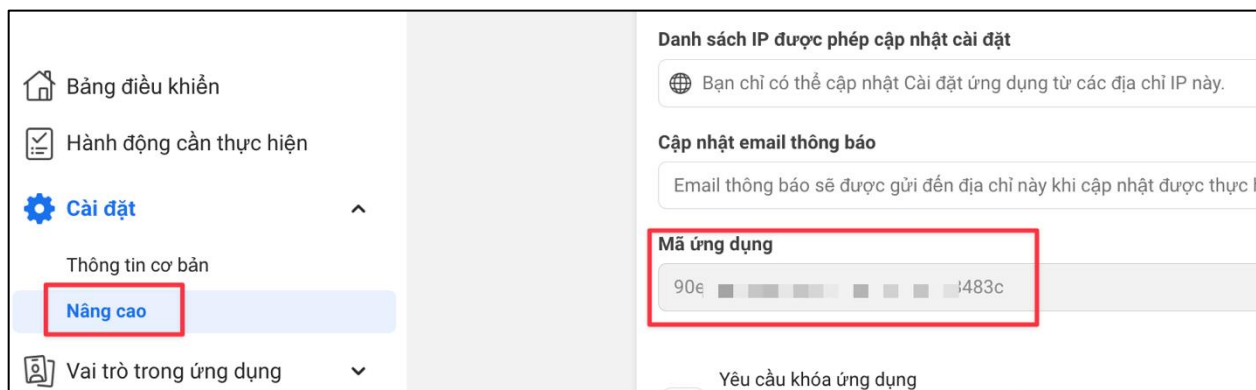
1 <resources>
2   <string name="app_name">Lab8</string>
3
4   <string name="facebook_app_id">64439...9794</string>
5   <string name="fb_login_protocol_scheme">fb6443...9794</string>
6   <string name="facebook_client_token">27c76e1a044008...22cadd8</string>
7 </resources>
  
```

- **facebook\_app\_id** ta sẽ lấy ở mã ứng dụng ngay tại thanh bar của trang web



- **fb\_login\_protocol\_scheme** ta chỉ cần thêm chữ “fb” vào trước *facebook\_app\_id* là được ví dụ : fb37102....28698
- **facebook\_client\_token**

Vào Cài đặt → Nâng Cao → Mã ứng dụng



Tiếp theo, trong AndroidManifest ta thêm 2 meta-data trong thẻ application

```
<meta-data
    android:name="com.facebook.sdk.ApplicationId"
    android:value="@string/facebook_app_id" />
<meta-data
    android:name="com.facebook.sdk.ClientToken"
    android:value="@string/facebook_client_token" />
```

```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="Lab8"
    android:supportsRtl="true"
    android:theme="@style/Theme.Lab1"
    tools:targetApi="31">
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="${MAPS_API_KEY}" />

    <meta-data
        android:name="com.facebook.sdk.ApplicationId"
        android:value="6447794" />
    <meta-data
        android:name="com.facebook.sdk.ClientToken"
        android:value="27c76e1a044222cadd8" />
```

Sau đó tiếp tục thêm vào 2 activity (do thư viện cung cấp)

```
<activity
    android:name="com.facebook.FacebookActivity"

    android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation" />

<activity
    android:name="com.facebook.CustomTabActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data android:scheme="@string/fb_login_protocol_scheme" />
    </intent-filter>
</activity>
```

```
<meta-data
    android:name="com.facebook.sdk.ApplicationId"
    android:value="64439 [REDACTED] 794" />

<meta-data
    android:name="com.facebook.sdk.ClientToken"
    android:value="27c76e1a04 [REDACTED] 222cadd8" />

<activity
    android:name="com.facebook.FacebookActivity"
    android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation" />

<activity
    android:name="com.facebook.CustomTabActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data android:scheme="fb6443 [REDACTED] 794" />
    </intent-filter>
</activity>
```

Xin thêm quyền internet cho ứng dụng

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
```

Trong Activity gọi hàm **onActivityResult** để nhận kết quả trả về sau khi đăng nhập

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    callbackManager.onActivityResult(requestCode, resultCode, data);
    super.onActivityResult(requestCode, resultCode, data);
}
```

Viết thêm hàm **getUserProfile** để lấy thông tin tài khoản đang đăng nhập

```
private void getUserProfile(AccessToken accessToken) {
    GraphRequest request = GraphRequest.newMeRequest(accessToken, new GraphRequest.GraphJSONObjectCallback() {
        @Override
        public void onCompleted(JSONObject object, GraphResponse response) {
            try {
                String name = object.getString( name: "name");
                String email = object.getString( name: "id");
                String image = object.getJSONObject( name: "picture").getJSONObject( name: "data").getString( name: "url");
                Toast.makeText( context: HomeActivity.this, text: name + " - " + email + " - " + image, Toast.LENGTH_SHORT).show();
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });
    Bundle parameters = new Bundle();
    parameters.putString("fields", "id,name,email,picture.width(200)");
    request.setParameters(parameters);
    request.executeAsync();
}
```

Trong onCreate viết đoạn code sau:

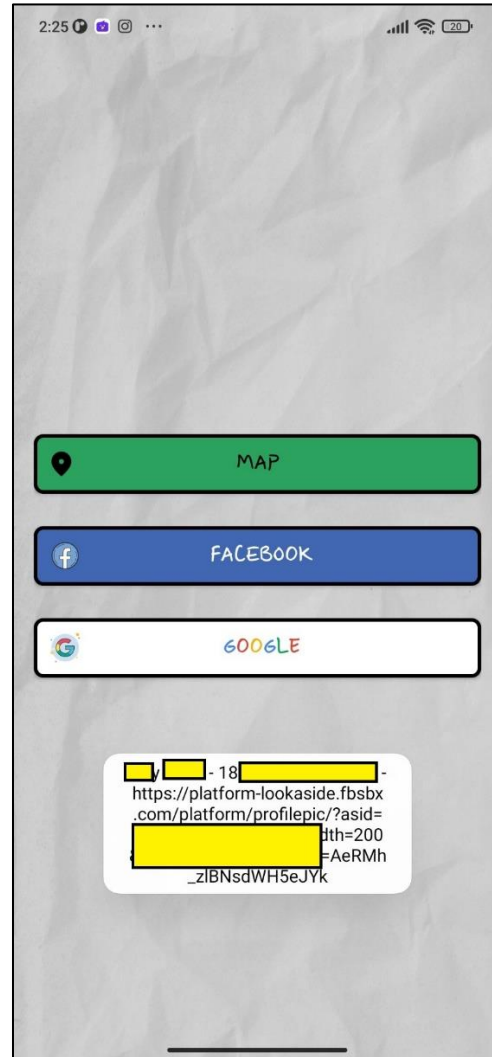
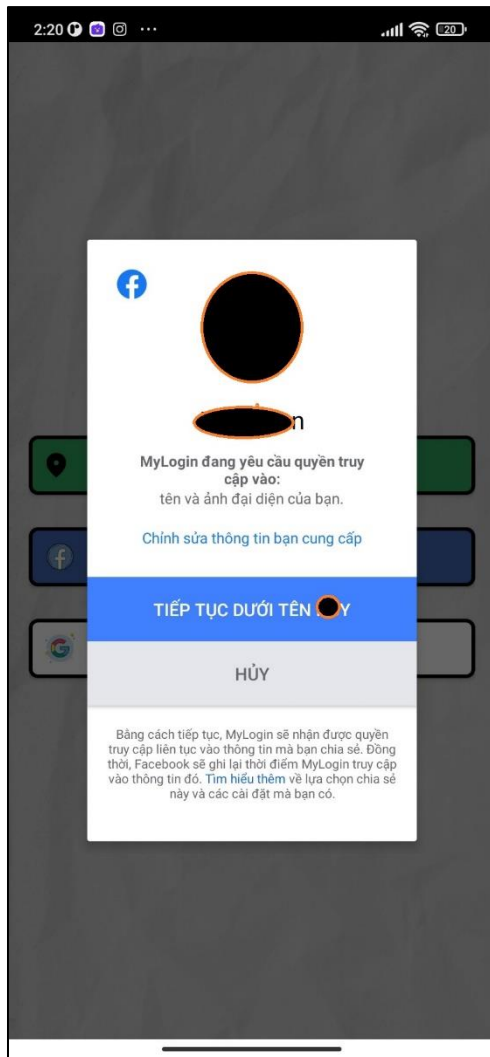
```
callbackManager = CallbackManager.Factory.create();
LoginManager.getInstance().registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) {
        // App code
        getUserProfile(loginResult.getAccessToken());
    }

    @Override
    public void onCancel() {
        // App code
        Toast.makeText(context: HomeActivity.this, text: "Đăng nhập không thành công", Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onError(FacebookException exception) {
        // App code
    }
});

btnFacebook.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        LoginManager.getInstance().loginWithReadPermissions(activity: HomeActivity.this, Arrays.asList("public_profile"));
    }
});
```

Ta được Kết quả:





## Thông tin thêm

1. Đăng xuất tài khoản ta chỉ cần ta gọi `LoginManager.getInstance().logout();`
2. Tạo nút đăng nhập do thư viện cung cấp sẵn

```
<com.facebook.login.widget.LoginButton
    android:id="@+id/login_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="30dp"
    android:layout_marginBottom="30dp" />
```



Trong phần xử lý code chúng ta có thể gọi trực tiếp `callbackManager` ở trong nút login thư viện cung cấp sẵn

```
loginButton = (LoginButton) findViewById(R.id.login_button);
loginButton.setReadPermissions(Arrays.asList("public_profile"));
// If you are using in a fragment, call loginButton.setFragment(this);

// Callback registration
loginButton.registerCallback(callbackManager, new
FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) {
        // App code
    }

    @Override
    public void onCancel() {
        // App code
    }

    @Override
    public void onError(FacebookException exception) {
        // App code
    }
});
```

#### **BÀI 4: GV CHO THÊM**

##### **\*\*\* YÊU CẦU NỘP BÀI:**

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết ---